Homework 7

The purpose of this homework is to practice wrangling data and conducting inference for simple linear regression models. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday November 3rd.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Data wrangling with dplyr

On July 3rd 2015, my 1999 Toyota Corolla broke down on the side of the highway outside of Sturbridge MA. While I had the car repaired, I knew it was time to sell it and get a new car. I intended to sell my Corolla to the car dealership, the only catch was that I was not sure how much the used Corolla was worth. In the following excercises we will model how much a used Corolla is worth as a function of the number of miles it has been driven.

The data we will look at comes from Edmunds.com which is a website where you can buy new and used cars online. This data set is from the 2015 DataFest competition, which is an undegraduate data science competition that takes place at difference colleges across the United States. The data has been made available to this class for educational purposes, however please do not share this data outside of the class.

Part 1.1 (15 points): Let's start by loading the dplyr libary and data set using the code below. Report how many cases and variables the full data set has. Then use the dplyr select() and filter() functions to create a reduced data frame object called `used_corollas` in which:

1) The only variables in that should be in the `used_corollas` data frame are:

a) model_bought: the model of the car
b) new_or_used_bought: whether a car was new or used when it was purchased
c) price_bought: the price the car was purchased for
d) mileage_bought: the number of miles the car had when it was purchased

2) The only cases should be in the `used_corollas` data frame are:

a) used cars
b) Toyota Corollas
c) cars that have been drive less than 150,000 miles

3) Finally use the na.omit() function on the `used_corollas` data frame to remove cases that have missing values.

If you have properly filter the data, the resulting data set should have 248 cases, so check this is the case before going on to the next set of exercises.

```
# load the dplyr library
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# load the data set
load("car_transactions.rda")


# get the size of the original data set
dim(car_transactions)
```

```
## [1] 107832     21
```

```
#glimpse(car_transactions)

#use dplyr to reduce the data set to only used Corolla's with under 150,000 miles
used_corollas <- filter(car_transactions, model_bought == "Corolla",
                        new_or_used_bought == "U", mileage_bought < 150000)
used_corollas <- select(used_corollas, model_bought,
                        new_or_used_bought, mileage_bought, price_bought)
used_corollas <- na.omit(used_corollas)

# check the size of the resulting data frame
dim(used_corollas)
```

```
## [1] 248   4
```

Answers

There are 21 variables and 107832 observations in the original data set. There are 248 cases and 4 variables in the filtered data set.

Part 2: Fitting a linear model and statistical inference on regression coefficients

Now that we have the relevant data, let's examine the relationship between a car's price and the number of miles driven!

Part 2.1 (10 points): Let's begin analyzing the data by taking the following steps:

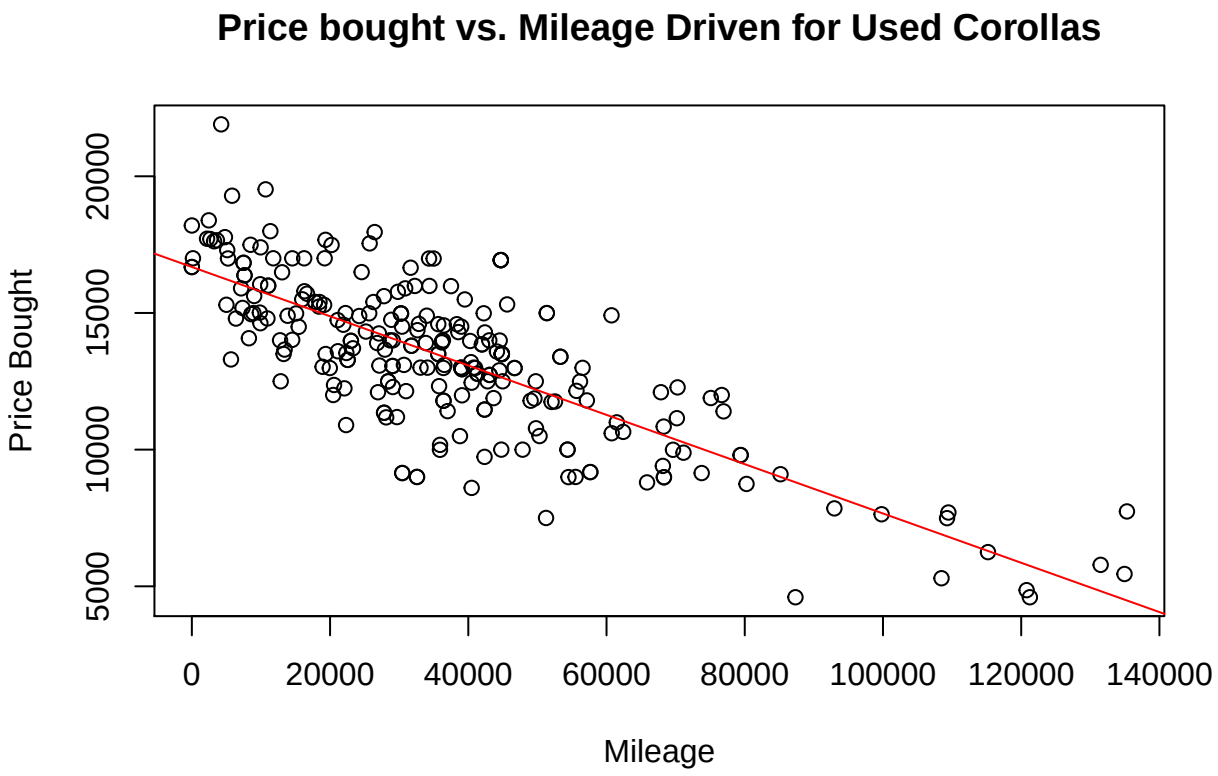
1) Plot the price as a function of the number of miles driven.
2) Fit a linear model regression model that shows the predicted (expected) price as a function of the number of miles driven. Save this model to an object called `lm_fit` which you will use throughout the rest of this homework.
3) Add a red line to our plot showing the regression line fit.
4) Print the regression coefficients found.

Report how much does the price of a Corolla decrease for every additional mile it has been driven, and also what this regression model suggests a car that has been driven 0 miles would be worth. Finally, write out the regression equation.

```
# let's start by plotting the data
plot(used_corollas$mileage_bought, used_corollas$price_bought,
     main = "Price bought vs. Mileage Driven for Used Corollas",
     xlab = "Mileage", ylab = "Price Bought")


# fit a regression model  (note: this is y as as function of x)
lm_fit <- lm(price_bought ~ mileage_bought, data = used_corollas)

# add the regression line to the plot
abline(lm_fit, col = "red")
```




```
# print the regression coefficients
coef(lm_fit)
```

```
##    (Intercept) mileage_bought
## 16681.91992781    -0.09018627
```

Answers:

The price of a Corolla decreases −$0.09 for every additional mile driven. The model suggests a used car at 0 miles would be worth $16681.92.

The regression equation is:

x = miles driven

$\hat{y}$ = predicted price of Corolla

$\hat{y} = -0.0902x + 16681.92$

Part 2.2 (5 points): Now use R's summary() function to report whether there is statistically significant evidence that the price of a car decreases as a function of the number of miles driven. Also, write out the hypothesis that is being tested using the appropriate symbols/notation discussed in class.

```
# get information about the statistical significance of the fit
summary(lm_fit)
```

```
##
## Call:
## lm(formula = price_bought ~ mileage_bought, data = used_corollas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4791.8 -1131.9    -0.3  1027.7  5600.7
##
## Coefficients:
##                    Estimate   Std. Error t value            Pr(>|t|)
## (Intercept)    16681.919928   204.459353   81.59 <0.0000000000000002 ***
## mileage_bought    -0.090186     0.004539  -19.87 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1816 on 246 degrees of freedom
## Multiple R-squared:  0.616,  Adjusted R-squared:  0.6145
## F-statistic: 394.7 on 1 and 246 DF,  p-value: < 0.00000000000000022
```

```
(p_value <- summary(lm_fit)$coefficients[2,4])
```

```
## [1] 0.00000000000000000000000000000000000000000000000000004747174
```

Answer

$H_0 : \beta_1 = 0$

$H_A : \beta_1 \neq 0$

$\alpha = 0.05$

The p-value is very close to 0 and less than our signficance level, so we reject the null hypothesis. There is statistically significant evidence of correlation between mileage and price.

Part 2.3 (5 points): We can create confidence intervals using a t-distribution via the confint() function. Report what the confidence interval for slope of the regression line is. Also, based on the confidence interval, explain why it it seems likely that the price of a car is not independent of the number of miles driven.

```
confint(lm_fit)
```

```
##                        2.5 %         97.5 %
## (Intercept)    16279.20570876 17084.63414685
## mileage_bought    -0.09912747    -0.08124508
```

Answer

The confidence interval is [-0.09912747, -0.08124508]. Since this interval does not include 0 which would imply no correlation (and we know it has a 95% probability of containing the true parameter), it seems likely that the price of a car is not independent of the number of miles driven.

Part 2.4 (10 points): We can also use the bootstrap to create confidence intervals for the slope of the regression coefficient. To do this you can use the following procedure:

1) Create a bootstrap resampled data frame by sampling with replacement from the `used_corollas` data frame. You can do this using dplyr's sample_n() function with the sample size being the number of cases in the `used_corollas` data frame and setting the `replace = TRUE` argument.

2) Fit the regression model using the bootstrap data frame.

3) Extract the regression slope coefficient and save it to a vector object.

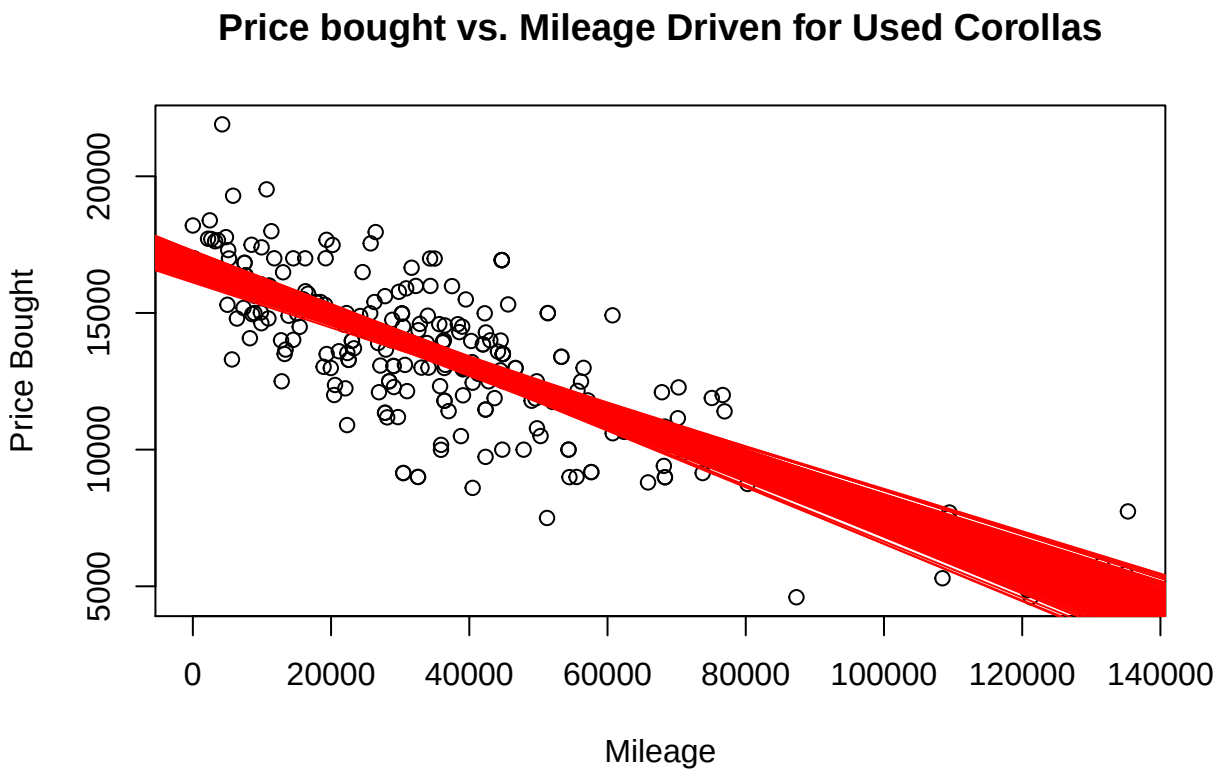
4) Repeat this process 1,000 times (this is less than normal because it is computationally expensive).

5) Use the percentile method to report a 95% confidence interval for the regression slope.

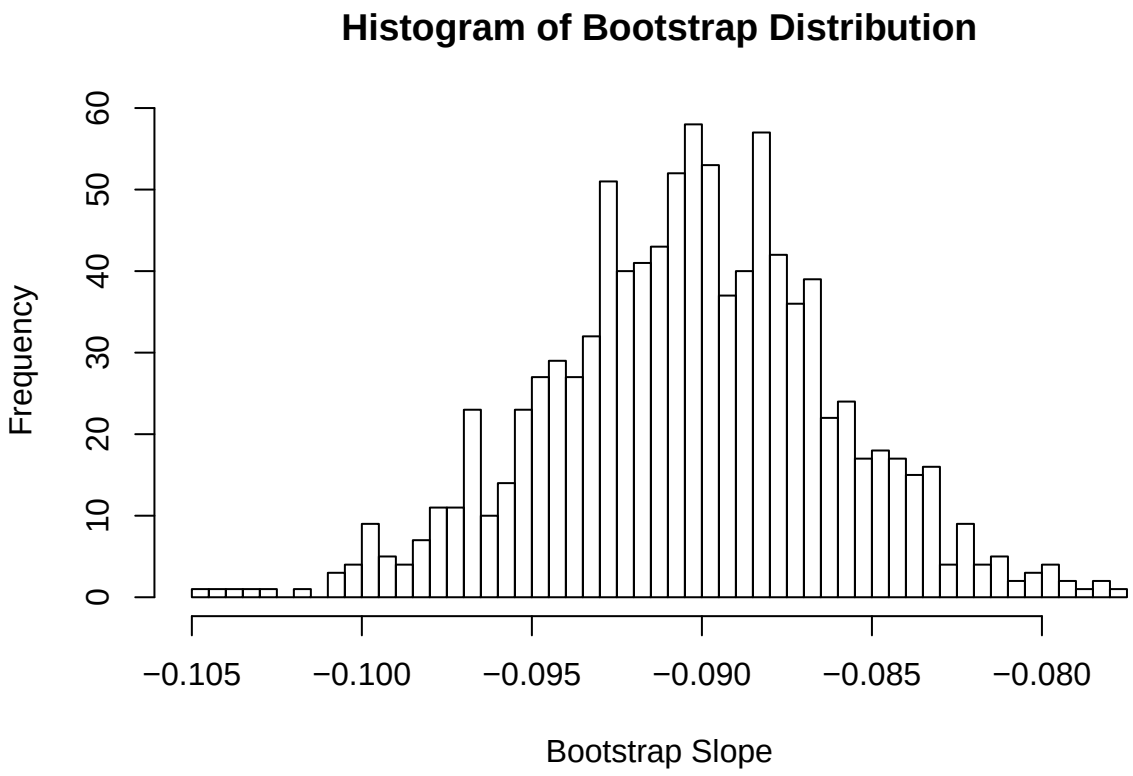
Also report whether the bootstrap confidence interval is similar to the confidence interval using the t-distribution you calculated above.

```
plot(used_corollas$mileage_bought, used_corollas$price_bought,
     main = "Price bought vs. Mileage Driven for Used Corollas",
     xlab = "Mileage", ylab = "Price Bought")

# create the bootstrap distribution
nrep <- 1000
n_cases <- dim(used_corollas)[1]
result_vec <- rep(0, nrep)
for (i in 1:nrep){
  boot_sample <- dplyr::sample_n(used_corollas, size = n_cases, replace = TRUE)
  boot_fit <- lm(price_bought ~ mileage_bought, data = boot_sample)
  abline(boot_fit, col = "red")
  result_vec[i] <- coef(boot_fit)[2]
}
```




```r
# plot it
hist(result_vec, main = "Histogram of Bootstrap Distribution",
     xlab = "Bootstrap Slope", ylab = "Frequency", nclass = 50)
```




```
# create bootstrap confidence intervals
boot_conf <- quantile(result_vec, c(0.025, 0.975))
boot_conf
```

```
##        2.5%       97.5%
## -0.09905251 -0.08216317
```

Answers:

The 95% Confidence interval is [-0.0990525, -0.0821632]. It is similar to the one calculated using the t-distribution, and both do not include 0.

Part 2.5 (10 points): My Toyota has 180,000 miles at the time I wanted to sell it. Based on the regression model fit above, what is the predicted worth of this car? Does this seem like a reasonable estimate?

```
#Worth of car calculated using predict() function
predict(lm_fit, newdata = data.frame(mileage_bought = 180000))
```

```
##        1
## 448.3911
```

```
#Worth of the car calculated using the above regression equation
16681.91993 - 0.09019*180000
```

```
## [1] 447.7199
```

Answer The predicted worth of the car using the predict() function is $448.39. I would say this estimate is a bit too low since even if the mileage is high a car would rarely be sold for so little. It might be possible to sell at this price since the car has already broken down on the highway before, but based on other prices online, a similar car should go for aroun $2000 or more. The low estimate is probably due to extrapolation from the line since the mileage we are given in our dataset does not exceed 140000.

Part 3: Analysis of variance (ANOVA) for regression

As discussed in class, we can also use an ANOVA to test the regression coefficients. We will explore relationship between the ANOVA and other analysis methods below.

Part 3.1 (5 points): Let's look at the relationship between the ANOVA F-statistic and the t-statistic. Use the anova() function on the linear model you fit and print out the ANOVA table. Look back to question 2.2 and create an object called `t_stat` that has the t-value that you was obtained from using the summary() function to get the t-statistic for the regression slope. Show that this the value of `t_stat` squared is (approximately) equal to the F-statistic found with the anova() function by printing both the t^2 value and the F-statistic value. Also, report the value of the sum of the model sum of squares (SSModel) and the residual sum of squares (SSError).

```
anova(lm_fit)
```

```
## Analysis of Variance Table
##
## Response: price_bought
##                  Df     Sum Sq    Mean Sq F value                Pr(>F)
## mileage_bought    1 1302085889 1302085889   394.7 < 0.00000000000000022 ***
## Residuals       246  811530941    3298906
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#T-value squared from summary function
t_stat <- -19.86712
(t_stat <- t_stat^2)
```

```
## [1] 394.7025
```

```
#F-statistic
anova(lm_fit)$"F value"[1]
```

```
## [1] 394.7023
```

```r
#SSModel
(ssmodel <- anova(lm_fit)$"Sum Sq"[1])
```

```
## [1] 1302085889
```

```r
#SSError
(sserror <- anova(lm_fit)$"Sum Sq"[2])
```

```
## [1] 811530941
```

```r
(sstotal <- ssmodel + sserror)
```

```
## [1] 2113616830
```

Answers:
As shown above, the t-value squared and F-statistic are approximately equal (394.7025 and 394.7023 respectively). SSModel is 1302085889.42664 and SSError is 811530940.763768. SSTotal is 2113616830.1904.

Part 3.2 (10 points): We can also extract the SSModel, SSError and SSTotal using values from the original data and from values stored in the `lm_fit` object. Run the following analyses to to calculate the SSModel, SSError and SSTotal values:

1) For the SSTotal, use the `used_corolla` data frame to calculate $\Sigma_{i=1}^{n}(y_i - \bar{y})^2$.
2) Use the `fitted.values` in the `lm_fit` object to caluclate SSModel using the formula $\Sigma_{i=1}^{n}(\hat{y}_i - \bar{y})^2$.
3) Use the `residuals` in the `lm_fit` object to caluclate SSError using the formula $\Sigma_{i=1}^{n}(\hat{y}_i - y_i)^2$.

To check you have the right answers, look at the values you got in question 3.1. Show that SSTotal = SSModel + SSError for the values you calculated.

```r
# the total sum of squares
(SSTotal <- (sum((used_corollas$price_bought - mean(used_corollas$price_bought))^2)))
```

```
## [1] 2113616830
```

```r
# the model sum of squares
(SSModel <- (sum((lm_fit$fitted.values - mean(used_corollas$price_bought))^2)))
```

```
## [1] 1302085889
```

```r
# the sum of squared error
(SSError <- sum(lm_fit$residuals^2))
```

```
## [1] 811530941
```

```
# show that SSTotal is equal to SSModel + SSError
SSError + SSModel
```

```
## [1] 2113616830
```

```
SSTotal
```

```
## [1] 2113616830
```

As we can see SSTotal = SSModel + SSError.

Part 3.3 (5 points): We also discussed in class that for simple linear regression, correlation coefficient squared (r^2) is equal to the percentage of the variance explained by the liner model: SSModel/SSTotal. Calculate the correlation coefficient between mileage_bought and price_bought, and square it (which gives you the *coefficient of determination*). Then using the values calculated in part 3.2, show that this is equal to SSModel/SSTotal, and also equal to 1 - SSResidual/SSTotal.

```
(coefdet <- (cor(used_corollas$mileage_bought, used_corollas$price_bought))^2)
```

```
## [1] 0.6160463
```

```
SSModel/SSTotal
```

```
## [1] 0.6160463
```

```
1 - SSError/SSTotal
```

```
## [1] 0.6160463
```

As we can see these are all equal.

Part 4: Regression diagnostics

When making inferences about regression coefficients, there are a number of assumptions that need to be met to make these tests/confidence invervals valid. The assumptions for an ANOVA are:

1) **Normality**: residuals are normally distributed around the predicted value $\hat{y}$
2) **Homoscedasticity**: constant variance over the whole range of x values
3) **Linearity**: A line can describe the relationship between x and y
4) **Independence**: each data point is independent from the other points

To check whether these assumptions seem to be met by creating a set of diagnostic plots.

Part 4.1 (5 points): To check whether the residuals are normally distributed we can use create a Q-Q plot. The `car` package has a nice function to create these plots called qqPlot() to create these plots. If we pass the `lm_fit` object to the qqPlot() function it will create a Q-Q plot of the studentized residuals. Create this plot and report if the residuals seem normally distributed?

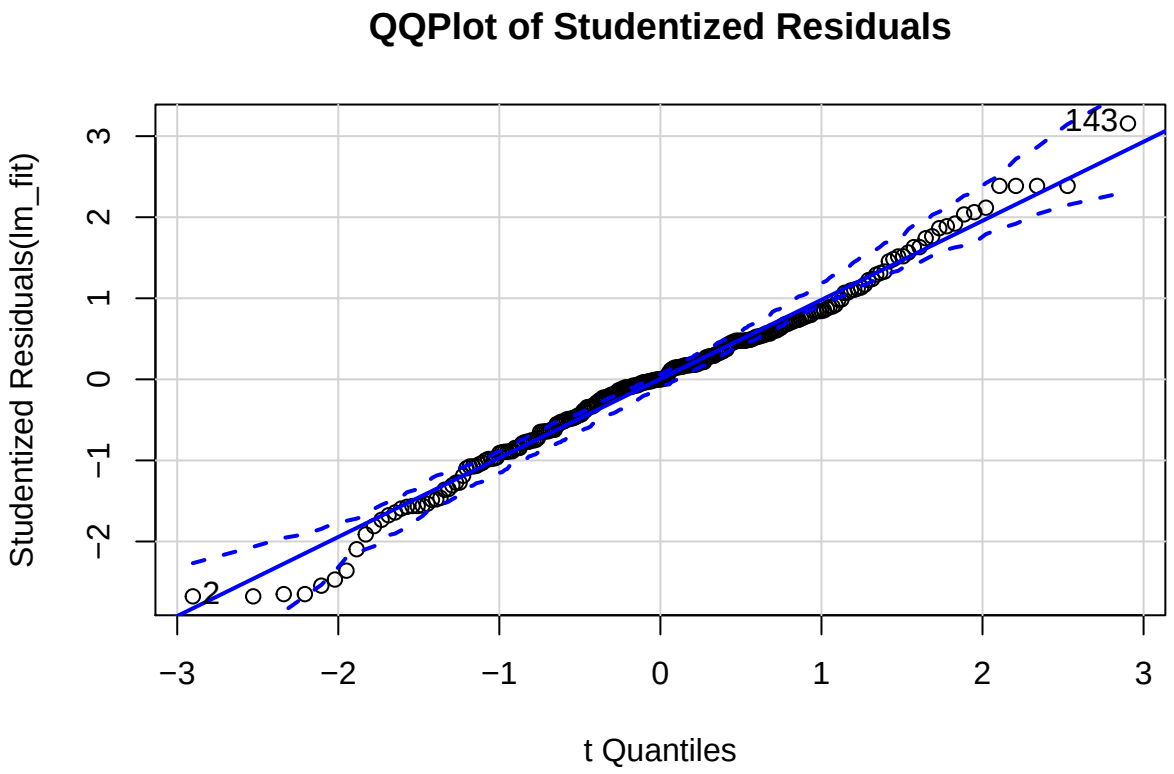
```
# install.packages('car')
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
qqPlot(lm_fit, main = "QQPlot of Studentized Residuals")
```



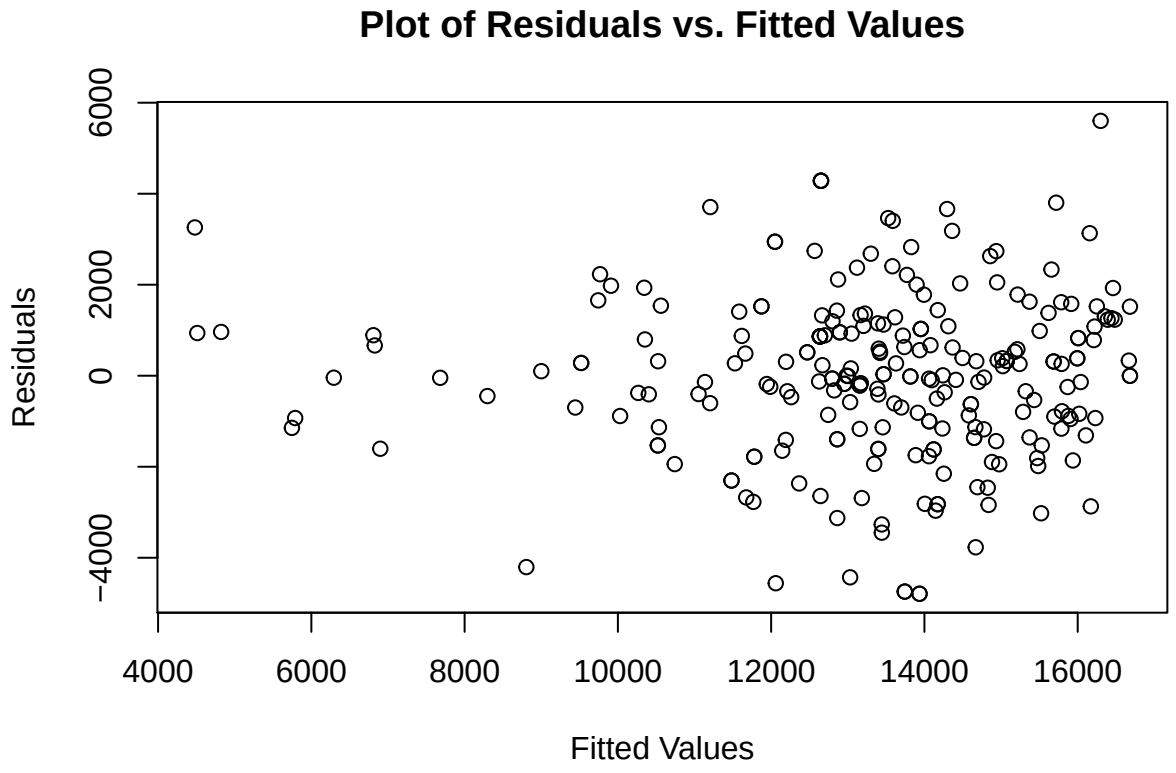

```
##   2 143
##   2 140
```

Answer:

The residuals are very close to the linear line y = x despite some variation at either end of the line, so they appear to be normally distributed.

Part 4.2 (5 points): To check for homoscedasticity and linearity, we can create a plot of the residuals as a function of the fitted values. Create such a plot below using information in the lm_fit object. Does it appear that homoscedasticity and linearity are met here? Are these results what you would expect from looking at plots above and from the nature of the type of data you are analyzing?

```
plot(lm_fit$fitted.values, lm_fit$residuals,
     main = "Plot of Residuals vs. Fitted Values",
     xlab = "Fitted Values", ylab = "Residuals")
```



Answers:

It is difficult to tell here, but it seems that the residuals seem to vary more as the fitted values increase, which shows that the plot exhibits some heteroscedasticity. This might be because cars with higher mileages all depreciate in a similar way, whereas cars with lower mileages may carry more varying values depending on other features of the car.

However, the residuals also seem to be centered around 0 with no particular pattern suggesting non-linearity. So based off of this plot I would say the linear model still stands. I originally thought that since the price of a used car should not go below 0, using a linear model to fit the data may not be the best choice - instead, a model like an inverse graph may be better if we want to predict the price of cars with very high mileage.

But based on the data frame we are given, where the mileage is still relatively low, a linear model still seems to work.

Part 4.3 (5 points): To check if the data points are independent requires knowledge of how the data was collected. For example, if the data you have is from a time-series (e.g., recordings of the temperature in New Haven on consecutive days) then there is a high likelihood that the data points might not be independent. On the other hand, if you take a simple random sample from a population where every point is equally likely to be selected, then the data is going to be independent.

Unfortunately I do not know exactly how this data was collected so it is difficult to say if the data is independent here. However, there might be ways to investigate whether it seems plausbile that it could be independent. Please describe some ways you might investigate whether the data could be independent (hint: think about the varaibles in the full `car_transactions` data set) Note: there is no exact 'right answer' here, just describe some possible ideas.

Answer:

To investigate whether or not the data is independent we could look at the dates and locations the cars were sold (date_sold and dma_bought, or another indicator of date and location). It is possible that some particular dealerships might be offering higher or lower prices on average than others, and if many cars in the dataset we are given were all sold at the same dealership, this may be reason to question the independence of the data points. The same goes for the state in which the car was sold. Similarly, perhaps during a certain period in time the value of a particular make/model of a car was increased or decreased, which would affect the independence of the data. For example, cars usually sell for higher around the holidays but then drop in price after the new year.

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 7